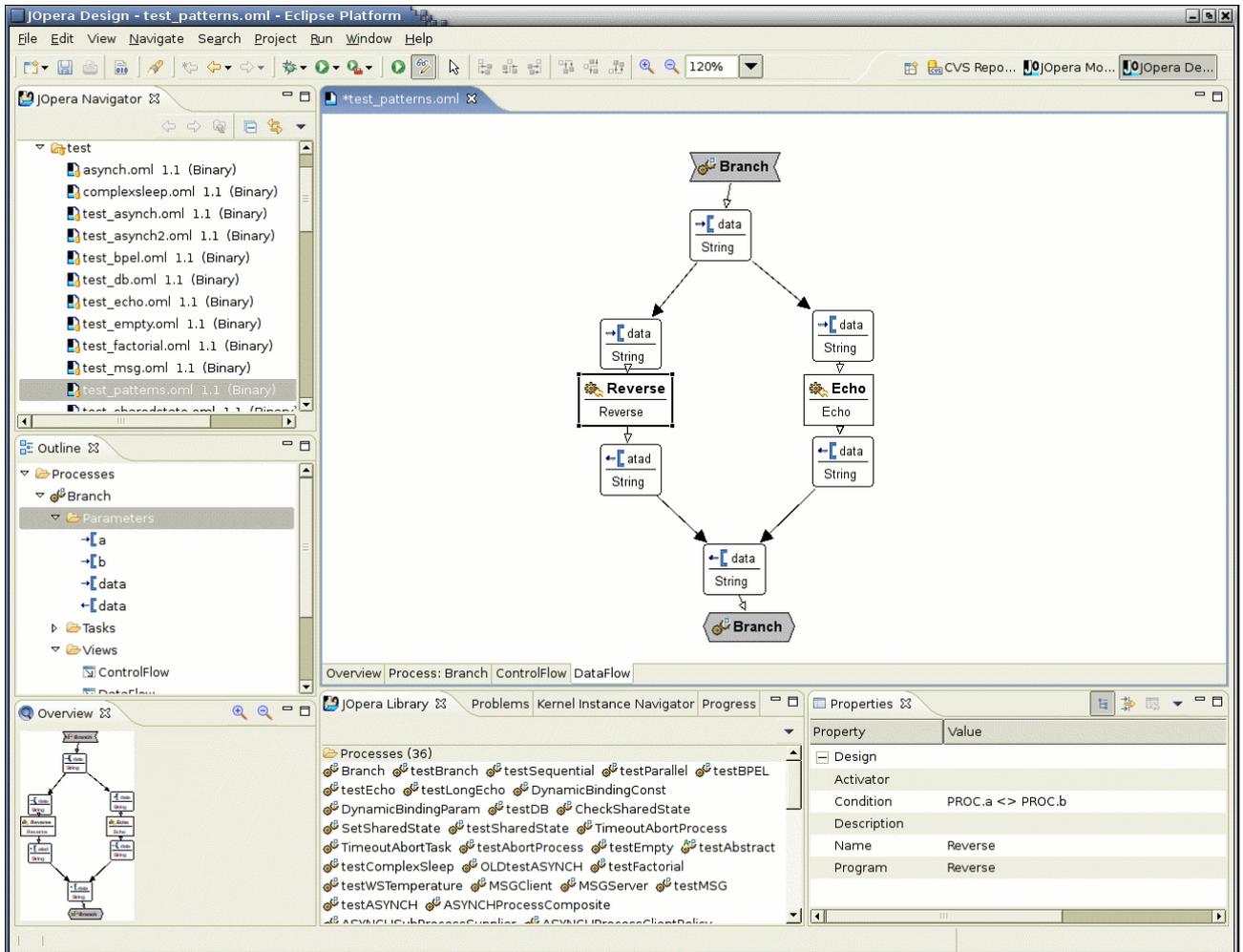


JOpera

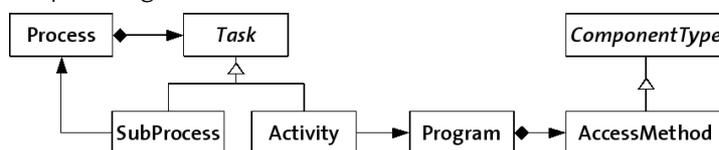
A Flexible System for Visual Service Composition

Andreas Bur – bura@student.ethz.ch

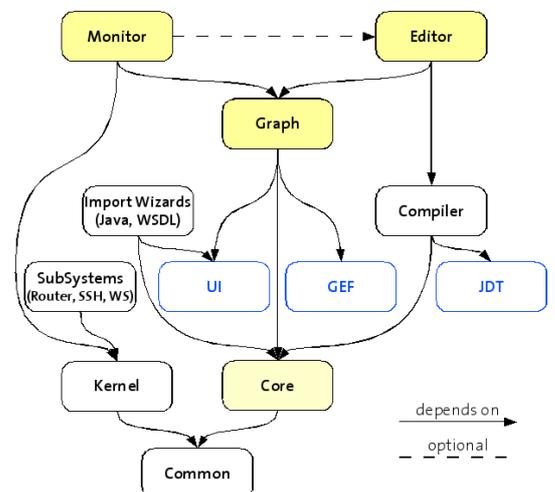


JOpera ist ein Projekt der Forschungsgruppe für Informations- und Kommunikationssysteme am Departement für Informatik der ETH Zürich. Es wird geleitet von Dr. Cesare Pautasso. Ziel ist es, ein einfach zu bedienendes System zur Komposition neuer Services bestehend aus vorhandenen Einzelkomponenten zu erstellen, die dann in einem Grid bestehend aus JOpera Systemen abgearbeitet werden. Dies ermöglicht zum einen, Nicht-Informatikern einen einfachen Zugang zu einem Hochleistungssystem und zum andern vereinfacht es ganz allgemein den Zusammenbau bestehender Systeme, wie dies zum Beispiel im Bereich der Enterprise Application Integration eine grundlegende Aufgabe ist.

JOpera besteht im wesentlichen aus zwei Teilen, einer Workflow Engine genannt Kernel und der graphischen Benutzeroberfläche. Beide Komponenten sind in Java implementiert, was einen plattformübergreifenden Einsatz ermöglicht. Die Benutzeroberfläche liegt als eine Reihe von Eclipse-Plugins vor.



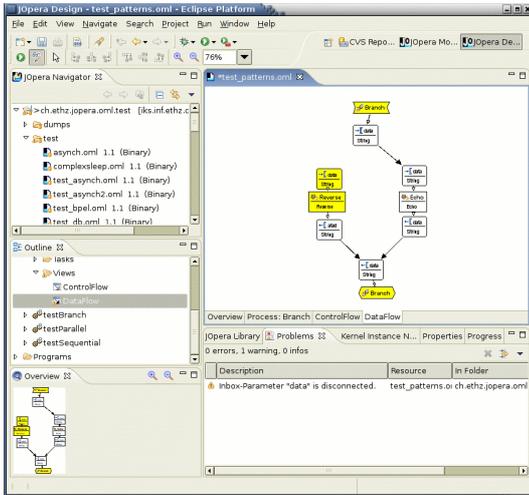
Prozessstruktur (vereinfacht)



Plugin-Struktur (vereinfacht)
blau: vorhandene Eclipse Plugins

Kernel

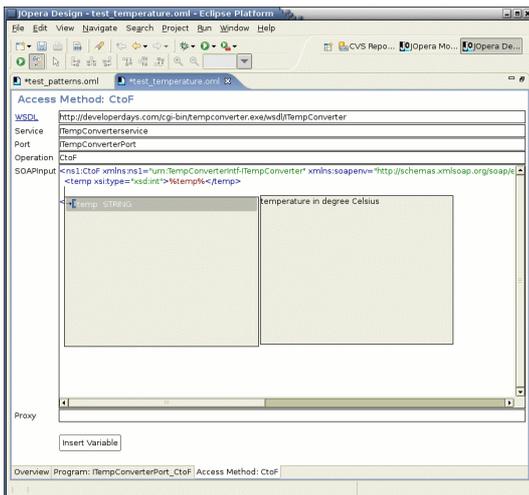
Der Kernel funktioniert als eine Workflow Engine, welche die verschiedenen Tasks gemäss ihren Abhängigkeiten abarbeitet. Sein Aufbau ist sehr modular. Dies ermöglicht das Ausführen unterschiedlichster Tasks, begonnen vom einfachen UNIX Kommando, über den Austausch von SOAP Nachrichten bis hin zum Ausführen von Java Code.



Färbung des Editors bei Fehlern (hier: disconnected Inbox-Parameter)

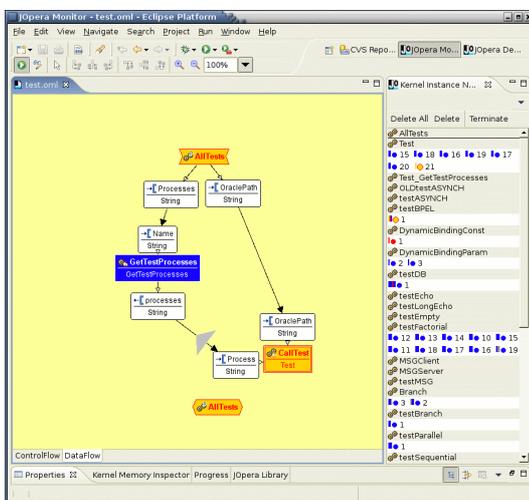
Userinterface

Das UI ist nahtlos in die vorhandene Eclipse-Infrastruktur eingebettet und erweitert diese wo nötig. In JOpera werden Prozesse in der JOpera Visual Composition Language (JVCL) definiert. Diese graphische Sprache erlaubt es innert kürzester Zeit bestehende Komponenten zu einem neuen Service zu verbinden. Basierend auf dem GEF Toolkit aus dem Eclipse Tools Projekt stellt JOpera einen graphischen Editor zur Verfügung, der sowohl zum Erstellen von Prozessen, wie auch zur Überwachung im Kernel laufender Prozesse gebraucht wird. Er ist mit den verschiedenen Standard-Views der Workbench verknüpft (Outline View, Problems View, Properties View). Per Drag&Drop lassen sich Tasks aus den verschiedenen Views spielend leicht einfügen. Probleme werden dem Benutzer durch entsprechende Färbung der Box signalisiert und ein Marker wird in die Problems View eingefügt; falls vorhanden mit den dazu passenden Quick Fixes.



Syntax Coloring und Auto Completion im dynamisch erstellten AccessMethod-Editor

Der graphische Editor wird durch eine Reihe von Forms-basierten Seiten ergänzt. Während diese bei Prozessen das Setzen von Metadaten erlauben, sind sie für Programme und AccessMethods die zentrale Editierumgebung. Wie in der vereinfachten Prozessstruktur (Vorderseite) angedeutet, ist die Logik der einzelnen Komponenten in AccessMethods integriert. Deren Felder variieren je nach Typ des zugrundeliegenden ComponentTypes, also dem Interface zu den verschiedenen vom Kernel unterstützten Subsystemen (z.B.: UNIX, SOAP, JAVASCRIPT). Um dieser Modularität gerecht zu werden, wird das Aussehen des AccessMethod-Editors dynamisch basierend auf dem ComponentType erstellt. Innerhalb der verschiedenen Felder stehen dem Benutzer die von Eclipse gewohnten Features wie Syntax Coloring und Auto Completion zur Verfügung. Zusätzlich werden Felder, welche eine URI enthalten, mit einem Link ausgestattet, der das Ziel im standard System Browser öffnet.



Monitor mit Kernel Instance Navigator

Wie bereits erwähnt wird der graphische Editor nicht bloss zum Erstellen neuer Prozesse gebraucht, sondern auch um diese während ihrer Abarbeitung im Kernel zu überwachen. Um zum einen die Anzahl offener Editoren zu vermindern und zum andern dem Anwender eine einheitliche Sicht seiner Prozesse zu ermöglichen, ist die Unterstützung von verschiedenen Modes eingebaut worden. Diese ermöglichen es, ein- und denselben Graphen mit unterschiedlichen Funktionen auszustatten. Während dies im Edit-Mode die verschiedenen Werkzeuge zum Bearbeiten der Prozesse sind, sind es im Monitor-Mode vor allem Hilfsmittel, um den aktuellen Zustand eines Prozesses oder seiner Bestandteile abzufragen. Im Monitor-Mode werden die verschiedenen Boxen je nach ihrem aktuellen Status gefärbt und Zwischenresultate und Fehlermeldungen können von Tooltips oder mit Hilfe der Properties View abgelesen werden.

Link: <http://www.jopera.iks.ethz.ch/>